A History of Computers and Computing

Michael D. Godfrey
`michaeldgodfrey@gmail.com`

May 2013, update: April 2026

1 Introduction

2 Early: Babbage, Penn, Bletchley Park, Manchester

3 Middle: Usable machines

4 Recent: Universal Use

5 Summary

Introduction

Background

The purpose of this talk is to present key events (as known to me), in the development of computing since about 1946. I only started to learn about all this in 1956. For convenience the history is divided into four periods: NOTE: Babbage information added April 2024. And, see Additional References 1.

First:	mid-late 1800's:	Babbage designed and verified first computer
Early:	1940's – 1960:	re-discovery
Middle:	1960 – 2010:	technical and commercial development
Current:	2010 – present:	Universal usage

Definition

Computers transform data by the application of rules. A solution is reached when the process reaches equilibrium: the outputs are consistent with the inputs. All operations are carried out on bit sequences in such a manner that the process is **exactly** reproducible. Keep in mind: computers only understand bits, not what they mean.

Introduction

Origin of the Stored Program Idea

The question is often asked: “Who invented the computer?” Various answers have been given, but Babbage as described in his autobiography is definitely first. He also verified his design using his RTL language. The design verification was completed in mid-1800’s.

Sequence: the computer as an automated desk calculator → the machine can manipulate its own programs.

Eckert-Mauchly

- Engineering Group first built ENIAC (plug-board machine) then EDVAC. The EDVAC described in the von Neumann Report was a 32bit zero address design. The EDVAC that was built was a 44bit three address machine. As Harry Huskey said “I tried to read the report, but could not make head or tail of it.” Subequently the same group became the Eckert-Mauchly Co. and built the UNIVAC I, LARC, and subsequent UNIVAC systems.
- Von Neumann EDVAC Report introduced schematics using logic from Pitts and McCulloch (1943). This is not widely known since the reprinting of the report omitted the last 10 Chapters. (See *The Origins of Digital Computers*, ed. B. Randell, Springer-Verlag, 1973.)
- Mauchly visit with Atanasoff and Berry.
- The Atanasoff equation solver was not a computer. The Judge in Minneapolis was wrong. (I worked for Cliff Berry while at Caltech.)

Bletchley Park

- Turing, Newman, Good, Flowers.
- Bombes (100's, Enigma, 1939), Colossi (10, Lorenz, Tunny, 1946 D-day).
- Lorenz code breaking (Tiltman): lorenz/fish.htm Vernam additive code.

Manchester

- Williams and Kilburn in the EE Department. Max Newman in Math. (1945)

- Turing came later when he got fed up with NPL. (NPL computer (ACE) as finally built was considerably influenced by Harry Huskey from Eckert-Mauchly. No one in England seems to know or would admit this.)

- The “Baby” machine (SSEM) had been mostly designed and implemented when Turing arrived, but he played a key role in its programming and further development. After Kilburn and Tootill, he was the first programmer. Derrick Morris learned programming from Turing and became a main architect and programmer of the series of Manchester computers.

Manchester

- First “commercial” machines. Mercury, Atlas, MU5, MU6.
- These machines were “marketed” by Ferranti, which became ICL, up to MU5. At that point the commercial relationship broke down and MU6 was not developed into a product or even described in publications.
- Introduced memory protection and paging (On the London Atlas, paging was to bidirectional magnetic tape!).
- Introduced multiprogramming, remote access.
- Consistent filesystems.
- Initially, programming, above assembly level, was in Autocode. Fortran and other languages were provided on Atlas.

AT&T, Bell Labs

- 1948: Stibitz machine, introduced floating point and remote access.
- 1956: All computer research terminated. Stibitz moved to Dartmouth.
- 1965: ESS1 Trenton, NJ. No writable memory. 1975: ESS4
- The “babysitter” problem. No reset and no visible state.
- 1969: End of MULTICS, start of UNIX.
- 1970: International direct dial Manhattan – London.
- 1972: AT&T Policy Directive (Bob Santos): “There will be no operational use of UNIX within the Bell System.”
- 1982: Final Consent Decree. Divestiture and end of BTL.
- 2005: the end. (SBC)

IBM

- 1958–62: 7090/4
- 1964: 360 360/67 (“address translation” design borrowed from GE, invention licensed from Manchester). 360/91
- Kilburn visit to Armonk for paging patent license.
- 1970–72: 370 ... 370/168 370/195
- 2000: eserver zseries.
- 2010: zEnterprise Systems.

UNIVAC

- Eckert and Mauchly setup Eckert-Mauchly Computer Corp. (1946) First computer was EDVAC II, renamed UNIVAC I and sold to Census for 1950 Census.
- UNIVAC I,..., 1108 (1964), 1100/60 (1979), ... ClearPath Dorado
- RTL language (Cray)
- Precise interrupts
- Virtual file system
- Demonstrably correct systems: full logic through checking, containment of errors, automatic fault recovery. Later adopted by IBM.
- Others: Cray, DEC, Burroughs

Integration of Computing and Communications

- Alan Bennett: “there is no period so remote as the recent past.”
- Google
- Phones, tablets, etc. Lack of standards, security, recovery. Some similarities with the early days of telephony when at least about four phones, each connected to a separate phone company, were needed to reach most people.
- There is uncertainty about the role of the internet in society and specifically in education.
- Coursera and edX show promise, but they are not close to a stable and well-understood feature of education.
- In any case, the real value of bricks and mortar (and, of course marble) Universities seems to deserve review.

Key Problems

- Memory Management: Segments vs. Paging.
- System Control Software. Modularity. Security. (B1, etc.) No steady state.
- Hospital systems: patient deaths due to computer malfunction.
- Inconsistent databases (EXT, etc.). btrfs supports copy-write, but is still not consistent.
- Monotonic databases that work. Google talks about this, but...
- Disconnect between Academic Research and the world. The papers on silent errors [1], [2], [3], [4] have not been referred to, nor have they led to further study.

Today

- **How to make an unreliable system from reliable parts.**
- Current digital systems are unsafe even though many of the parts are very reliable.
- The path from here to safe and reliable computing and communications is very unclear.
- None of the organizations which could make a major contribution show much sign of interest.

References I

[1] BAIRAVASUNDARAM, L. N. & ARPACI-DUSSEAU, A. C. & ARPACI-DUSSEAU, R. H. & GOODSON, G. R. & SCHROEDER, B.
An Analysis of Data Corruption in the Storage Stack.
ACM Transactions on Storage (TOS) 4, 3 (Nov 2008), 8:2–8:28.
Available at: acm.org.

[2] BAIRAVASUNDARAM, L. N. & ARPACI-DUSSEAU, A. C. & ARPACI-DUSSEAU, R. H. & GOODSON, G. R. & SCHROEDER, B.
Corruption in the Storage Stack: a closer look.
The Usenix Magazine 33, 3 (Jun 2008), 6–14.
Available at: usenix.org.

References II

[3] PETERSEN, M. K.
I/O Controller Requirements for Data Integrity Aware Operating Systems.
Oracle Inc. (2008).
Available at: oss.oracle.com.

[4] PETERSEN, M. K.
I/O Controller Data Integrity Extensions.
Oracle Inc. (2009).
Available at: oss.oracle.com.

Additional References I

- Babbage, *Passages from the Life of a Philosopher,* Cambridge Library Collection, Cambridge Press, 2011.
- Copeland, *Colossus,* Oxford Press, 2006.
- Gordon Welchman, Herman Lukoff, von Neumann (Caltech Lecture), vN_Caltech_Lecture.pdf